|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **第四届浙江省大学生工程训练综合能力竞赛**  The 4th ZheJiang UndergraduateEngineering Training Integration Ability Competition | **电路设计方案**  Electronic Circuit Design Scheme | | 共 4 页 | 第 1 页 |
| 编 号 | （工作人员填写） |
|  | | **底盘运动模块设计方案**  **1.电路层面**  电路层面底盘主要分为驱动和寻迹两个模块。  驱动模块方面，由于工程实践与创新大赛的比赛场地是很规则的方形，且循迹的黑色轨迹为正交的直线，我们选择采用麦克纳姆四轮驱动进行跑图。使用麦轮运动的优点是它不需要任何旋转，只需平移即可到达任何目标地点，而从左侧示意图上看，平移则图上对角麦轮的有效速度是平行同向的，这就使本来四路使能端分别驱动四轮的电路可以简化为对角的电机共用一个使能端口，但是由于实际运动过程中存在误差，小车的实际转速差或打滑会导致小车旋转，因此需要将角速度分量叠加到小车上，因此仍然使用4个使能端口输出PWM波控制，每个接口另外两个数字信号引脚控制小车前进后退和停止。所以麦轮小车共占用8个数字输出端口和4个模拟输出端口。由于控制板不能直接提供驱动电机的电流，使用L298N芯片进行转接驱动，芯片最主要的是一个H桥电路结构如图所示，我们将使用将2个L298N芯片集成在一个主板上的模块。  底盘除了4个电机还有一个循迹传感器模块。我们使用四边分别7路灰度传感器的检测装置识别轨迹（没有找到其他适合尺寸的传感器）。在4边分别装配是为了减少车轮打滑或转速差造成的角度偏差。灰度传感器原理在这里不加赘述，实际使用只要知道它会返回给控制板一个范围内的值，表征反射光强度，我们可以根据实际情况调节各灰度的阈值使黑色返回1，其他颜色返回0（如果是数字传感器不需要人为设定阈值，直接返回0和1）。 为了节省I-O口，尽量使用串口与主板通信。  虽然用寻迹模块做闭环控制可以使小车在轨迹上运行，但为了保证稳定性，我们不能只依赖寻迹模块去调控小车的电机转速，由于4个电机参数有细微的差异，输入相同的电压时转速可能不同，何况输入电压值也难以保证相同，因此需要利用电机自带的编码器反馈一个实际转速的值，根据这个值用PID方式调节转速至目标转速，保证小车沿直线行进。  由于主控板可能引脚数不足，底盘的电机驱动模块暂定使用下位机，同上位机利用串口通信交互参数。  综上，底盘电路结构的框架如左图所示。  底盘各类输入输出端口占用数量统计：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 模块 | 占用数字I-O口 | 占用模拟输入口 | 占用PWM输出口 | 占用串口 | | 电机驱动 | 2\*4 | 0/4 | 1\*4 | 0 | | 电机编码器 | 2\*4 | 0 | 0 | 0 | | 寻迹模块 | 0 | 0 | 0 | 4 | | 总计 | 16 | 0/4 | 4 | 4 | | | |
| **2.控制算法层面**  首先是麦克纳姆轮的方向控制。基础功能是实现底盘在X-Y两个方向的平动，进阶功能是底盘绕中心旋转来调整方向，均涉及简单的逆运动学解算知识。如左图所示，我们暂且将前进方向作为X轴正方向，左侧作为Y轴正方向，我们所希望达到的目的是给定X-Y轴向速度分量，程序将其分解到各个电机的转速上去。麦克纳姆轮的特点是电机的角速度沿Y轴方向，但提供的线速度是沿与地面接触辊子的轴向方向，因此有效速度分量为与X-Y轴分别夹45°的方向。不妨如图所示设置有效速度的正方向。线速度大小要根据小车的实际尺寸和轮径算出，此处从略。要保证小车平移而不旋转，若相对于世界坐标系对角电机转动方向相反，则另外两个电机转向也只能相反，最后只会一动不动。因此对角轮一定是同时向前或者向后旋转，左上和右下轮提供y = -x直线方向的速度，同理右上和坐下轮提供y = x方向分量，则左上轮转速w = k(vx-vy)，同理可求得其他轮的转速。而要实现旋转只需要将角速度分解得到的转速分量和线速度的转速分量做线性叠加即可，注意顺时针和逆时针的正负号问题。  编码器读取电机转速只要在一定时间内读取脉冲数量乘一个比例系数即可，根据从A端口和B端口读到的脉冲相位差可以看出电机是顺时针还是逆时针旋转。  寻迹模块采用寻迹的PID算法，为每个传感器加上适当的速度权重，根据实际读到的值线性叠加实现变速回调。  为了使系统更快速而稳定地达到目标状态（角度和速度），电机调速采用PID算法，输入量为目标速度和电机编码器返回速度，输出PWM参数，利用J-Scope PID调试工具调整PID系数至最适。  以上均为机器人本身的微观控制层面的算法分析，为了实现跑图，下述宏观调整策略：  全局控制策略是分阶段实现控制，设置控制变量来判断此时应处于任务的哪个阶段。当小车出发时先循迹，读到目标个数的黑色横轨道(7个传感器全返回黑色对应数值)，停止进入下一个阶段。进行特定任务时配合视觉系统给出的参数调整小车的位姿。整个控制方式以闭环控制为主，在局部使用开环控制，如为了防止计数器重复技术，在读到横条给计数器计数+1的同时关闭计数器一段很小的时间段，保证小车越过横条再重新打开。为了防止每个阶段误差的累计，每个阶段任务完成后需要原地调整姿态。 | | |

学校名称：

装

订

线

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C:\Users\30465\Desktop\Automated_Guided_Vehicle\方案文档10.30\材料\速度分解.png | 关键代码说明  PID算法基本代码实现（图1）：  用SetTunings()函数为kp,ki,kd赋值，在中断服务函数或其回调函数里调用Compute()函数实现PID控速。  麦克纳姆轮速度解算（图2）：  GeneralSpeedToLinearSpeed()函数仅控制平动转速，RotateSpeedToLinearSpeed()函数叠加旋转速度分量。 | | | |
|  | | 关键代码说明 | 比 例 |  |
| **第四届浙江省大学生工程训练综合能力竞赛** | 共 页 | 第5 页 |

装

订

线

学校名称：